



Home | Login | Logout | Access Information | Alerts |
Welcome United States Patent and Trademark Office

Search Results

BROWSE

SEARCH

IEEE Xplore GUIDE

Results for "((memory compiler' and characteriz*)<in>metadata)"

Your search matched 3 of 1416205 documents.

A maximum of 100 results are displayed, 25 to a page, sorted by **Relevance in Descending** order.

e-mail

» Search Options

[View Session History](#)

[New Search](#)

Modify Search

((memory compiler' and characteriz*)<in>metadata)

Search

Check to search only within this results set

Display Format: Citation Citation & Abstract

» Key

IEEE JNL IEEE Journal or Magazine

IEE JNL IEE Journal or Magazine

IEEE CNF IEEE Conference Proceeding

IEE CNF IEE Conference Proceeding

IEEE STD IEEE Standard

[View Selected Items](#) [Select All](#) [Deselect All](#)

1. **A multi-megabit memory compiler: tomorrow's IP [integrated processor]**
Xian-Quan Zhang; Tsang, T.; Mehta, D.;
[Electrical and Computer Engineering, 1999 IEEE Canadian Conference on](#)
Volume 1, 9-12 May 1999 Page(s):538 - 542 vol.1
Digital Object Identifier 10.1109/CCECE.1999.807256
[AbstractPlus](#) | Full Text: [PDF\(496 KB\)](#) [IEEE CNF](#)
[Rights and Permissions](#)

2. **A diffused CMOS SRAM compiler for gate-arrays**

Gee, P.; Tou, J.;
[Circuits and Systems, 1991., Proceedings of the 34th Midwest Symposium on](#)
14-17 May 1991 Page(s):807 - 810 vol.2
Digital Object Identifier 10.1109/MWSCAS.1991.251990

[AbstractPlus](#) | Full Text: [PDF\(304 KB\)](#) [IEEE CNF](#)
[Rights and Permissions](#)

3. **On the yield of compiler-based eSRAMs**

Wang, X.; Ottavi, M.; Meyer, F.; Lombardi, F.;
[Defect and Fault Tolerance in VLSI Systems, 2004. DFT 2004. Proceedings. 1](#)
[International Symposium on](#)
10-13 Oct. 2004 Page(s):11 - 19
Digital Object Identifier 10.1109/DFTVS.2004.1347820

[AbstractPlus](#) | Full Text: [PDF\(330 KB\)](#) [IEEE CNF](#)
[Rights and Permissions](#)

[Help](#) [Contact Us](#) [Privacy &](#)

© Copyright 2006 IEEE -

Indexed by
 Inspec®

[Home](#) | [Login](#) | [Logout](#) | [Access Information](#) | [Alerts](#) |

Welcome United States Patent and Trademark Office

Search Results[BROWSE](#)[SEARCH](#)[IEEE XPLOR GUIDE](#) [e-mail](#)

Results for "((memory compiler' and characteriz* and interpolat*)<in>metadata)"

Your search matched **0** documents.A maximum of **100** results are displayed, **25** to a page, sorted by **Relevance** in **Descending** order.**» Search Options**[View Session History](#)**Modify Search**[New Search](#) Check to search only within this results setDisplay Format: Citation Citation & Abstract**» Key**

IEEE JNL IEEE Journal or Magazine

IEE JNL IEE Journal or Magazine

IEEE CNF IEEE Conference Proceeding

IEE CNF IEE Conference Proceeding

IEEE STD IEEE Standard

No results were found.

Please edit your search criteria and try again. Refer to the Help pages if you need assistance.

[Help](#) [Contact Us](#) [Privacy &](#)

© Copyright 2006 IEEE -

Indexed by
 Inspec®

[Home](#) | [Login](#) | [Logout](#) | [Access Information](#) | [Alerts](#) |

Welcome United States Patent and Trademark Office

 [Search Results](#)[BROWSE](#)[SEARCH](#)[IEEE Xplore GUIDE](#)

Results for "((memory compiler' and characteriz* and samp!*)<in>metadata)"

 [e-mail](#)

Your search matched 0 documents.

A maximum of 100 results are displayed, 25 to a page, sorted by **Relevance** in **Descending** order.**» Search Options**[View Session History](#)[Modify Search](#)[New Search](#) Check to search only within this results setDisplay Format: Citation Citation & Abstract**» Key**

IEEE JNL IEEE Journal or Magazine

IEE JNL IEE Journal or Magazine

IEEE CNF IEEE Conference Proceeding

IEE CNF IEE Conference Proceeding

IEEE STD IEEE Standard

No results were found.

Please edit your search criteria and try again. Refer to the Help pages if you need assistance.

[Help](#) [Contact Us](#) [Privacy &](#)

© Copyright 2006 IEEE -

Indexed by
 Inspec®



Home | Login | Logout | Access Information | Alerts |
Welcome United States Patent and Trademark Office

Search Results

BROWSE

SEARCH

IEEE Xplore GUIDE

Results for "((memory compiler' and characteriz* and 'scale factor')<in>metadata)"

Your search matched 0 documents.

A maximum of 100 results are displayed, 25 to a page, sorted by **Relevance in Descending** order.

e-mail

» [Search Options](#)

[View Session History](#)

[Modify Search](#)

[New Search](#)

((memory compiler' and characteriz* and 'scale factor')<in>metadata)

Search

Check to search only within this results set

Display Format: Citation Citation & Abstract

» **Key**

IEEE JNL IEEE Journal or Magazine

IEE JNL IEE Journal or Magazine

IEEE CNF IEEE Conference Proceeding

IEE CNF IEE Conference Proceeding

IEEE STD IEEE Standard

No results were found.

Please edit your search criteria and try again. Refer to the Help pages if you need assistance.

[Help](#) [Contact Us](#) [Privacy &](#)

© Copyright 2006 IEEE -

Indexed by
 Inspec®



Subscribe (Full Service) Register (Limited Service, Free) Login
Search: The ACM Digital Library The Guide
+ "memory compiler" +characteriz* +"scale factor"

Nothing Found

Your search for **+"memory compiler" +characteriz* +"scale factor"** did not return any results.

You may want to try an [Advanced Search](#) for additional options.

Please review the [Quick Tips](#) below or for more information see the [Search Tips](#).

Quick Tips

- Enter your search terms in lower case with a space between the terms.

sales offices

You can also enter a full question or concept in plain language.

Where are the sales offices?

- Capitalize proper nouns to search for specific people, places, or products.

John Colter, Netscape Navigator

- Enclose a phrase in double quotes to search for that exact phrase.

"museum of natural history" "museum of modern art"

- Narrow your searches by using a **+** if a search term must appear on a page.

museum +art

- Exclude pages by using a **-** if a search term must not appear on a page.

museum -Paris

Combine these techniques to create a specific search query. The better your description of the information you want, the more relevant your results will be.

museum +"natural history" dinosaur -Chicago

 [Subscribe \(Full Service\)](#) [Register \(Limited Service, Free\)](#) [Login](#)

Search: The ACM Digital Library The Guide

+"memory compiler" +characteriz* +interpolat*

SEARCH



 [Feedback](#) [Report a problem](#) [Satisfaction survey](#)

Terms used **memory compiler characteriz interpolat**

Found 1 of 185,942

Sort results by [Save results to a Binder](#)

[Try an Advanced Search](#)
[Try this search in The ACM Guide](#)

Display results [Search Tips](#)

[Open results in a new window](#)

Results 1 - 1 of 1

Relevance scale 

1 [Graph contraction for physical optimization methods: a quality-cost tradeoff for mapping data on parallel computers](#)



N. Mansour, R. Ponnusamy, A. Choudhary, G. C. Fox

August 1993 **Proceedings of the 7th international conference on Supercomputing**

Publisher: ACM Press

Full text available:  [pdf\(733.23 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

Mapping data to parallel computers aims at minimizing the execution time of the associated application. However, it can take an unacceptable amount of time in comparison with the execution time of the application if the size of the problem is large. In this paper, first we motivate the case for graph contraction as a means for reducing the problem size. We restrict our discussion to applications where the problem domain can be described using a graph (e.g., computational fluid dynamics appl ...

Results 1 - 1 of 1

The ACM Portal is published by the Association for Computing Machinery. Copyright © 2006 ACM, Inc.

[Terms of Usage](#) [Privacy Policy](#) [Code of Ethics](#) [Contact Us](#)

Useful downloads:  [Adobe Acrobat](#)  [QuickTime](#)  [Windows Media Player](#)  [Real Player](#)


[Subscribe \(Full Service\)](#) [Register \(Limited Service, Free\)](#) [Login](#)
[Search: The ACM Digital Library](#) [The Guide](#)

[THE ACM DIGITAL LIBRARY](#)
[Feedback](#) [Report a problem](#) [Satisfaction survey](#)
Terms used [memory compiler](#) [characteriz](#) [sampl](#)
Found 15 of 185,942
Sort results by
[Try an Advanced Search](#)
[Try this search in The ACM Guide](#)
Display results
 [Open results in a new window](#)
Results 1 - 15 of 15
Relevance scale

1 [Designers' forum: low power design: PowerViP: Soc power estimation framework at transaction level](#)



Ikhwan Lee, Hyunsuk Kim, Peng Yang, Sungjoo Yoo, Eui-Young Chung, Kyu-Myung Choi, Jeong-Taek Kong, Soo-Kwan Eo

January 2006 **Proceedings of the 2006 conference on Asia South Pacific design automation ASP-DAC '06**

Publisher: ACM Press

Full text available: [pdf\(349.97 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#)

In this work, we propose a SoC power estimation framework built on our system-level simulation environment. Our framework provides designers with the system-level power profile in a cycle-accurate manner. We target the framework to run fast and accurately, which is enabled by adopting different modeling techniques depending on the power characteristics of various IP blocks. The framework can be applied to any target SoC design.

2 [Energy-aware design of embedded memories: A survey of technologies, architectures, and optimization techniques](#)



Luca Benini, Alberto Macii, Massimo Poncino

February 2003 **ACM Transactions on Embedded Computing Systems (TECS)**, Volume 2 Issue 1

Publisher: ACM Press

Full text available: [pdf\(288.44 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

Embedded systems are often designed under stringent energy consumption budgets, to limit heat generation and battery size. Since memory systems consume a significant amount of energy to store and to forward data, it is then imperative to balance power consumption and performance in memory system design. Contemporary system design focuses on the trade-off between performance and energy consumption in processing and storage units, as well as in their interconnections. Although memory design is as ...

Keywords: Embedded systems, embedded memories, integration, memories, nonvolatile, system-on-a-chip, volatile

3 [A pipelined memory architecture for high throughput network processors](#)



Timothy Sherwood, George Varghese, Brad Calder

May 2003 **ACM SIGARCH Computer Architecture News , Proceedings of the 30th**

annual international symposium on Computer architecture ISCA '03, Volume 31 Issue 2

Publisher: ACM Press

Full text available: [pdf\(213.66 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#)

Designing ASICs for each new generation of backbone routers is a time intensive and fiscally draining process. In this paper we focus on the design of a programmable architecture for backbone routers, based on the manipulation of wide irregular memory words, that can provide a feasible design alternative to custom ASICs. We propose a pipelined memory design that emphasizes worst-case throughput over latency, and co-explore architectural tradeoffs with the design of several important network algo ...

4 Compiler Managed Dynamic Instruction Placement in a Low-Power Code Cache

Rajiv A. Ravindran, Pracheeti D. Nagarkar, Ganesh S. Dasika, Eric D. Marsman, Robert M. Senger, Scott A. Mahlke, Richard B. Brown

March 2005 **Proceedings of the international symposium on Code generation and optimization CGO '05**

Publisher: IEEE Computer Society

Full text available: [pdf\(631.68 KB\)](#) Additional Information: [full citation](#), [abstract](#), [index terms](#)

Modern embedded microprocessors use low power on-chip memories called scratch-pad memories to store frequently executed instructions and data. Unlike traditional caches, scratch-pad memories lack the complex tag checking and comparison logic, thereby proving to be efficient in area and power. In this work, we focus on exploiting scratch-pad memories for storing hot code segments within an application. Static placement techniques focus on placing the most frequently executed portions of programs ...

5 Security processor design: Power estimation starategies for a low-power security

processor

Yen-Fong Lee, Shi-Yu Huang, Sheng-Yu Hsu, I-Ling Chen, Cheng-Tao Shieh, Jian-Cheng Lin, Shih-Chieh Chang

January 2005 **Proceedings of the 2005 conference on Asia South Pacific design automation ASP-DAC '05**

Publisher: ACM Press

Full text available: [pdf\(398.87 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#)

In this paper, we present the power estimation methodologies for the development of a low-power security processor that contains significant amount of logic and memory. For the logic part, we present a highly accurate tool, called *PowerMixer*. This tool is a refinement of the so-called mixed-level methodology that combines the accuracy of quick SPICE and the speed of gate-level simulation. A grouping scheme is proposed so as to improve the accuracy for design blocks as large as 100K gates. ...

6 Online Only: ACM Transactions on Design Automation of Electronic Systems, vol. 11, issue 3 (Novel Paradigms in System-Level Design): Architecture description

language (ADL)-driven software toolkit generation for architectural exploration of programmable SOCs

Prabhat Mishra, Aviral Srivastava, Nikil Dutt

June 2004 **ACM Transactions on Design Automation of Electronic Systems (TODAES) , Proceedings of the 41st annual conference on Design automation DAC '04**, Volume 11 Issue 3

Publisher: ACM Press

Full text available: [pdf\(1.07 MB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

Advances in semiconductor technology permit increasingly complex applications to be realized using programmable systems-on-chips (SOCs). Furthermore, shrinking time-to-market demands, coupled with the need for product versioning through software

modification of SOC platforms, have led to a significant increase in the software content of these SOCs. However, designer productivity is greatly hampered by the lack of automated software generation tools for the exploration and evaluation of different ...

Keywords: Architecture description language, design space exploration, embedded processor, programmable architecture, retargetable compilation

7 Compiling Fortran D for MIMD distributed-memory machines

 Seema Hiranandani, Ken Kennedy, Chau-Wen Tseng
August 1992 **Communications of the ACM**, Volume 35 Issue 8

Publisher: ACM Press

Full text available:  pdf(5.38 MB)

Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#), [review](#)



Keywords: Fortran D, concurrent languages, distributed languages, distributed programming, parallel languages, parallel programming

8 A general framework for prefetch scheduling in linked data structures and its application to multi-chain prefetching

 Seungryul Choi, Nicholas Kohout, Sumit Pamnani, Dongkeun Kim, Donald Yeung
May 2004 **ACM Transactions on Computer Systems (TOCS)**, Volume 22 Issue 2

Publisher: ACM Press

Full text available:  pdf(2.45 MB)

Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)



Pointer-chasing applications tend to traverse composite data structures consisting of multiple independent pointer chains. While the traversal of any single pointer chain leads to the serialization of memory operations, the traversal of independent pointer chains provides a source of memory parallelism. This article investigates exploiting such *interchain memory parallelism* for the purpose of memory latency tolerance, using a technique called *multi--chain prefetching*. Previous work ...

Keywords: Data prefetching, memory parallelism, pointer-chasing code

9 Efficient resolution of sparse indirections in data-parallel compilers

 Manuel Ujaldon, Emilio L. Zapata
July 1995 **Proceedings of the 9th international conference on Supercomputing**

Publisher: ACM Press

Full text available:  pdf(1.05 MB)

Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#)



10 Architecture-independent scientific programming in data parallel C: three case studies

 Philip J. Hatcher, Michael J. Quinn, Ray J. Anderson, Anthony J. Lapadula, Bradley K. Seavers, Andrew F. Bennett
August 1991 **Proceedings of the 1991 ACM/IEEE conference on Supercomputing**

Publisher: ACM Press

Full text available:  pdf(1.05 MB)

Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#)



11 Compiler optimizations for Fortran D on MIMD distributed-memory machines

Seema Hiranandani, Ken Kennedy, Chau-Wen Tseng

August 1991 **Proceedings of the 1991 ACM/IEEE conference on Supercomputing****Publisher:** ACM PressFull text available:  [pdf\(1.57 MB\)](#) Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#)**12 Improving data locality with loop transformations**

Kathryn S. McKinley, Steve Carr, Chau-Wen Tseng

July 1996 **ACM Transactions on Programming Languages and Systems (TOPLAS)**, Volume 18 Issue 4**Publisher:** ACM PressFull text available:  [pdf\(411.40 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#), [review](#)

In the past decade, processor speed has become significantly faster than memory speed. Small, fast cache memories are designed to overcome this discrepancy, but they are only effective when programs exhibit data locality. In this article, we present compiler optimizations to improve data locality based on a simple yet accurate cost model. The model computes both temporal and spatial reuse of cache lines to find desirable loop organizations ...

Keywords: Cache, compiler optimization, data locality, loop distribution, loop fusion, loop permutation, loop reversal, loop transformations, microprocessors, simulation

13 Unified compilation of Fortran 77D and 90D

Alok Choudhary, Geoffrey Fox, Seema Hiranandani, Ken Kennedy, Charles Koelbel, Sanjay Ranka, Chau-Wen Tseng

March 1993 **ACM Letters on Programming Languages and Systems (LOPLAS)**, Volume 2 Issue 1-4**Publisher:** ACM PressFull text available:  [pdf\(1.29 MB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#), [review](#)

We present a unified approach to compiling Fortran 77D and Fortran 90D programs for efficient execution of MIMD distributed-memory machines. The integrated Fortran D compiler relies on two key observations. First, array constructs may be scalarized into FORALL loops without loss of information. Second, loop fusion, partitioning, and sectioning optimizations are essential for both Fortran D dialects.

Keywords: Fortran D, parallel languages, parallel programming

14 An architecture for software-controlled data prefetching

Alexander C. Klaiber, Henry M. Levy

April 1991 **ACM SIGARCH Computer Architecture News, Proceedings of the 18th annual international symposium on Computer architecture ISCA '91**, Volume 19 Issue 3**Publisher:** ACM PressFull text available:  [pdf\(1.16 MB\)](#) Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#)**15****Evaluation of compiler optimizations for Fortran D on MIMD distributed memory machines**

Seema Hiranandani, Ken Kennedy, Chau-Wen Tseng
August 1992 **Proceedings of the 6th international conference on Supercomputing**

Publisher: ACM Press

Full text available: [pdf\(1.74 MB\)](#)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

The Fortran D compiler uses data decomposition specifications to automatically translate Fortran programs for execution on MIMD distributed-memory machines. This paper introduces and classifies a number of advanced optimizations needed to achieve acceptable performance; they are analyzed and empirically evaluated for stencil computations. Profitability formulas are derived for each optimization. Results show that exploiting parallelism for pipelined computations, reductions, and scans is vi ...

Results 1 - 15 of 15

The ACM Portal is published by the Association for Computing Machinery. Copyright © 2006 ACM, Inc.

[Terms of Usage](#) [Privacy Policy](#) [Code of Ethics](#) [Contact Us](#)

Useful downloads: [Adobe Acrobat](#) [QuickTime](#) [Windows Media Player](#) [Real Player](#)

 [Subscribe \(Full Service\)](#) [Register \(Limited Service, Free\)](#) [Login](#)
Search: The ACM Digital Library The Guide
 USPTO

 [Feedback](#) [Report a problem](#) [Satisfaction survey](#)

Terms used [memory compiler characterization](#) Found 37 of 185,942

Sort results by [Save results to a Binder](#) Try an [Advanced Search](#)
 Display results [Search Tips](#) Try this search in [The ACM Guide](#)
 [Open results in a new window](#)

Results 1 - 20 of 37

Result page: **1** [2](#) [next](#)

Relevance scale 

1 [Designers' forum: low power design: PowerViP: Soc power estimation framework at transaction level](#) 
 Ikhwan Lee, Hyunsuk Kim, Peng Yang, Sungjoo Yoo, Eui-Young Chung, Kyu-Myung Choi, Jeong-Taek Kong, Soo-Kwan Eo
 January 2006 **Proceedings of the 2006 conference on Asia South Pacific design automation ASP-DAC '06**
Publisher: ACM Press
 Full text available:  [pdf\(349.97 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#)
 In this work, we propose a SoC power estimation framework built on our system-level simulation environment. Our framework provides designers with the system-level power profile in a cycle-accurate manner. We target the framework to run fast and accurately, which is enabled by adopting different modeling techniques depending on the power characteristics of various IP blocks. The framework can be applied to any target SoC design.

2 [Compiler support for hybrid irregular accesses on multiccomputers](#) 
 Antonio Lain, Prithviraj Banerjee
 January 1996 **Proceedings of the 10th international conference on Supercomputing**
Publisher: ACM Press
 Full text available:  [pdf\(999.10 KB\)](#) Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#)

3 [Performance of distributed sparse Cholesky factorization with pre-scheduling](#) 
 S. Venugopal, V. K. Naik, J. Saltz
 December 1992 **Proceedings of the 1992 ACM/IEEE conference on Supercomputing**
Publisher: IEEE Computer Society Press
 Full text available:  [pdf\(978.77 KB\)](#) Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#)

4 [Make Your SoC Design a Winner: Select the Right Memory IP](#) 
 V. Ratford
 March 2002 **Proceedings of the conference on Design, automation and test in Europe**
Publisher: IEEE Computer Society
 Full text available:  [pdf\(49.24 KB\)](#) Additional Information:

[Publisher Site](#)[full citation](#)

5 [Energy-aware design of embedded memories: A survey of technologies, architectures, and optimization techniques](#)

 Luca Benini, Alberto Macii, Massimo Poncino

February 2003 **ACM Transactions on Embedded Computing Systems (TECS)**, Volume 2 Issue 1

Publisher: ACM Press

Full text available:  [pdf\(288.44 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

Embedded systems are often designed under stringent energy consumption budgets, to limit heat generation and battery size. Since memory systems consume a significant amount of energy to store and to forward data, it is then imperative to balance power consumption and performance in memory system design. Contemporary system design focuses on the trade-off between performance and energy consumption in processing and storage units, as well as in their interconnections. Although memory design is as ...

Keywords: Embedded systems, embedded memories, integration, memories, nonvolatile, system-on-a-chip, volatile

6 [A pipelined memory architecture for high throughput network processors](#)

 Timothy Sherwood, George Varghese, Brad Calder

May 2003 **ACM SIGARCH Computer Architecture News, Proceedings of the 30th annual international symposium on Computer architecture ISCA '03**, Volume 31 Issue 2

Publisher: ACM Press

Full text available:  [pdf\(213.66 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#)

Designing ASICs for each new generation of backbone routers is a time intensive and fiscally draining process. In this paper we focus on the design of a programmable architecture for backbone routers, based on the manipulation of wide irregular memory words, that can provide a feasible design alternative to custom ASICs. We propose a pipelined memory design that emphasizes worst-case throughput over latency, and co-explore architectural tradeoffs with the design of several important network algo ...

7 [Using Mobilize Power Management IP for Dynamic & Static Power Reduction in SoC at 130 nm](#)

Dan Hillman

March 2005 **Proceedings of the conference on Design, Automation and Test in Europe - Volume 3 DATE '05**

Publisher: IEEE Computer Society

Full text available:  [pdf\(149.06 KB\)](#) Additional Information: [full citation](#), [abstract](#), [index terms](#)

At 130 nm and 90 nm, power consumption (both dynamic and static) has become a barrier in the roadmap for SoC designs targeting battery powered, mobile applications. This paper presents the results of dynamic and static power reduction achieved implementing Tensilica's 32-bit Xtensa microprocessor core, using Virtual Silicon's Power Management IP. Independent voltage islands are created using Virtual Silicon's VIP PowerSaver standard cells by using voltage level shifting cells and voltage isolati ...

8 [Area-Performance Trade-offs in Tiled Dataflow Architectures](#)

Steven Swanson, Andrew Putnam, Martha Mercaldi, Ken Michelson, Andrew Petersen, Andrew Schwerin, Mark Oskin, Susan J. Eggers

June 2006 **Proceedings of the 33rd International Symposium on Computer**

Architecture ISCA '06**Publisher:** IEEE Computer SocietyFull text available:  [pdf\(487.22 KB\)](#) Additional Information: [full citation](#), [abstract](#)

Tiled architectures, such as RAW, SmartMemories, TRIPS, and WaveScalar, promise to address several issues facing conventional processors, including complexity, wire-delay, and performance. The basic premise of these architectures is that larger, higher-performance implementations can be constructed by replicating the basic tile across the chip. This paper explores the area-performance trade-offs when designing one such tiled architecture, WaveScalar. We use a synthesizable RTL model and cycle-le ...

Keywords: WaveScalar, Dataflow computing, ASIC, RTL**9 Speculative prefetching**

Y. Jégou, O. Temam

August 1993 **Proceedings of the 7th international conference on Supercomputing****Publisher:** ACM PressFull text available:  [pdf\(1.12 MB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

A hardware prefetching mechanism named Speculative Prefetching is proposed. This scheme detects vector accesses issued by a load/store instruction and prefetches the corresponding data. The scheme requires no software add-on, and in some cases it is more powerful than software techniques for identifying regular accesses. The tradeoffs related to its hardware implementation are extensively discussed in order to finely tune the mechanism. Experiments show that average memory ...

10 Cache miss equations: a compiler framework for analyzing and tuning memory behavior

Somnath Ghosh, Margaret Martonosi, Sharad Malik

July 1999 **ACM Transactions on Programming Languages and Systems (TOPLAS)**,
Volume 21 Issue 4**Publisher:** ACM PressFull text available:  [pdf\(548.18 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#), [review](#)

With the ever-widening performance gap between processors and main memory, cache memory, which is used to bridge this gap, is becoming more and more significant. Caches work well for programs that exhibit sufficient locality. Other programs, however, have reference patterns that fail to exploit the cache, thereby suffering heavily from high memory latency. In order to get high cache efficiency and achieve good program performance, efficient memory accessing behavior is necessary. In fact, f ...

Keywords: cache memories, compilation, optimization, program transformation**11 Architecture and Design of a High Performance SRAM for SoC Design**

Shobha Singh, Shamsi Azmi, Nutan Aarawal, Penaka Phani, Ansuman Rout

January 2002 **Proceedings of the 2002 conference on Asia South Pacific design automation/VLSI Design****Publisher:** IEEE Computer SocietyFull text available:  [pdf\(186.65 KB\)](#) Additional Information: [full citation](#), [abstract](#)
 [Publisher Site](#)

Critical issues in designing a high speed, low power static RAM in deep submicron technologies are described along with the design techniques used to overcome them. With

appropriate circuit partitioning, transistor sizing, choice of a suitable Sense Amplifier, a good resetting technique and judicious use of dual Vth transistors we have achieved a high speed memory without dissipating too much power. The Introduction gives the specifications of the memory that was our design target. In Section II, w ...

12 Session 11A: embedded tutorial: System and architecture-level power reduction of microprocessor-based communication and multi-media applications 

Lode Nachtergael, Vivek Tiwari, Nikil Dutt

November 2000 **Proceedings of the 2000 IEEE/ACM international conference on Computer-aided design**

Publisher: IEEE Press

Full text available:  [pdf\(34.99 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#)

Current microprocessor architectures become more and more dominated by the data access bottlenecks in the cache, system bus and main memory subsystems. These also have a major influence on the system (board-level) power consumption. In practice this means lower energy consumption for a given throughput requirement. In the booming domain of (largely embedded) cost-sensitive communication and multi-media applications, more and more implementations make use of microprocessor based platforms for flex ...

13 Security processor design: Power estimation starategies for a low-power security processor 

 Yen-Fong Lee, Shi-Yu Huang, Sheng-Yu Hsu, I-Ling Chen, Cheng-Tao Shieh, Jian-Cheng Lin, Shih-Chieh Chang

January 2005 **Proceedings of the 2005 conference on Asia South Pacific design automation ASP-DAC '05**

Publisher: ACM Press

Full text available:  [pdf\(398.87 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#)

In this paper, we present the power estimation methodologies for the development of a low-power security processor that contains significant amount of logic and memory. For the logic part, we present a highly accurate tool, called *PowerMixer*. This tool is a refinement of the so-called mixed-level methodology that combines the accuracy of quick SPICE and the speed of gate-level simulation. A grouping scheme is proposed so as to improve the accuracy for design blocks as large as 100K gates. ...

14 PARADIGM: a compiler for automatic data distribution on multicomputers 

 Manish Gupta, Prithviraj Banerjee

August 1993 **Proceedings of the 7th international conference on Supercomputing**

Publisher: ACM Press

Full text available:  [pdf\(1.11 MB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

One of the most challenging steps in developing a parallel program for a distributed memory machine is determining how data should be distributed across processors. Most of the compilers being developed to make it easier to program such machines still provide no assistance to the programmer in this difficult and machine-dependent task. We have developed PARADIGM, a compiler that makes data partitioning decisions for Fortran 77 procedures. A significant feature of the design of PARADIGM is t ...

15 A general framework for prefetch scheduling in linked data structures and its application to multi-chain prefetching 

 Seungryul Choi, Nicholas Kohout, Sumit Pamnani, Dongkeun Kim, Donald Yeung

May 2004 **ACM Transactions on Computer Systems (TOCS)**, Volume 22 Issue 2

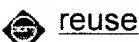
Publisher: ACM Press

Full text available: [pdf\(2.45 MB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

Pointer-chasing applications tend to traverse composite data structures consisting of multiple independent pointer chains. While the traversal of any single pointer chain leads to the serialization of memory operations, the traversal of independent pointer chains provides a source of memory parallelism. This article investigates exploiting such *interchain memory parallelism* for the purpose of memory latency tolerance, using a technique called *multi--chain prefetching*. Previous work ...

Keywords: Data prefetching, memory parallelism, pointer-chasing code

16 Runtime compilation techniques for data partitioning and communication schedule 



R. Ponnusamy, J. Saltz, A. Choudhary

December 1993 **Proceedings of the 1993 ACM/IEEE conference on Supercomputing**

Publisher: ACM Press

Full text available: [pdf\(967.75 KB\)](#) Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#)

17 Compiler Managed Dynamic Instruction Placement in a Low-Power Code Cache 

Rajiv A. Ravindran, Pracheeti D. Nagarkar, Ganesh S. Dasika, Eric D. Marsman, Robert M. Senger, Scott A. Mahlke, Richard B. Brown

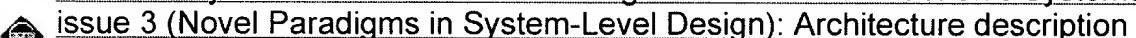
March 2005 **Proceedings of the international symposium on Code generation and optimization CGO '05**

Publisher: IEEE Computer Society

Full text available: [pdf\(631.68 KB\)](#) Additional Information: [full citation](#), [abstract](#), [index terms](#)

Modern embedded microprocessors use low power on-chip memories called scratch-pad memories to store frequently executed instructions and data. Unlike traditional caches, scratch-pad memories lack the complex tag checking and comparison logic, thereby proving to be efficient in area and power. In this work, we focus on exploiting scratch-pad memories for storing hot code segments within an application. Static placement techniques focus on placing the most frequently executed portions of programs ...

18 Online Only: ACM Transactions on Design Automation of Electronic Systems, vol. 11, 



issue 3 (Novel Paradigms in System-Level Design): Architecture description language (ADL)-driven software toolkit generation for architectural exploration of programmable SOCs

Prabhat Mishra, Aviral Srivastava, Nikil Dutt

June 2004 **ACM Transactions on Design Automation of Electronic Systems (TODAES) , Proceedings of the 41st annual conference on Design automation DAC '04**, Volume 11 Issue 3

Publisher: ACM Press

Full text available: [pdf\(1.07 MB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

Advances in semiconductor technology permit increasingly complex applications to be realized using programmable systems-on-chips (SOCs). Furthermore, shrinking time-to-market demands, coupled with the need for product versioning through software modification of SOC platforms, have led to a significant increase in the software content of these SOCs. However, designer productivity is greatly hampered by the lack of automated software generation tools for the exploration and evaluation of different ...

Keywords: Architecture description language, design space exploration, embedded processor, programmable architecture, retargetable compilation

19 Compiling for shared-memory and message-passing computers  James R. LarusMarch 1993 **ACM Letters on Programming Languages and Systems (LOPLAS)**, Volume 2
Issue 1-4**Publisher:** ACM PressFull text available:  [pdf\(1.27 MB\)](#)Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

Many parallel languages presume a shared address space in which any portion of a computation can access any datum. Some parallel computers directly support this abstraction with hardware shared memory. Other computers provide distinct (per-processor) address spaces and communication mechanisms on which software can construct a shared address space. Since programmers have difficulty explicitly managing address spaces, there is considerable interest in compiler support for shared address spaces ...

Keywords: cache coherence, compilers, directory protocols, memory systems, message-passing multiprocessors, parallel programming languages, shared-memory multiprocessors

20 Compiling Fortran D for MIMD distributed-memory machines  Seema Hiranandani, Ken Kennedy, Chau-Wen TsengAugust 1992 **Communications of the ACM**, Volume 35 Issue 8**Publisher:** ACM PressFull text available:  [pdf\(5.38 MB\)](#)Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#), [review](#)

Keywords: Fortran D, concurrent languages, distributed languages, distributed programming, parallel languages, parallel programming

Results 1 - 20 of 37

Result page: [1](#) [2](#) [next](#)

The ACM Portal is published by the Association for Computing Machinery. Copyright © 2006 ACM, Inc.

[Terms of Usage](#) [Privacy Policy](#) [Code of Ethics](#) [Contact Us](#)

Useful downloads:  [Adobe Acrobat](#)  [QuickTime](#)  [Windows Media Player](#)  [Real Player](#)

 [Subscribe \(Full Service\)](#) [Register \(Limited Service, Free\)](#) [Login](#)

Search: The ACM Digital Library The Guide

+memory compiler" +characterization

 [Feedback](#) [Report a problem](#) [Satisfaction survey](#)

Terms used [memory compiler characterization](#)

Found 37 of 185,942

Sort results by [Save results to a Binder](#)
 [Search Tips](#) [Open results in a new window](#)

Display results

[Try an Advanced Search](#)
[Try this search in The ACM Guide](#)

Results 21 - 37 of 37

Result page: [previous](#) [1](#) [2](#)

Relevance scale 

21 Compiler optimizations for Fortran D on MIMD distributed-memory machines 

 Seema Hiranandani, Ken Kennedy, Chau-Wen Tseng
 August 1991 **Proceedings of the 1991 ACM/IEEE conference on Supercomputing**

Publisher: ACM Press

Full text available:  [pdf\(1.57 MB\)](#) Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#)

22 Compiler-directed page coloring for multiprocessors 

 Edouard Bugnion, Jennifer M. Anderson, Todd C. Mowry, Mendel Rosenblum, Monica S. Lam
 September 1996 **ACM SIGPLAN Notices , ACM SIGOPS Operating Systems Review , Proceedings of the seventh international conference on Architectural support for programming languages and operating systems ASPLOS-VII**, Volume 31 , 30 Issue 9 , 5

Publisher: ACM Press

Full text available:  [pdf\(1.37 MB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

This paper presents a new technique, *compiler-directed page coloring*, that eliminates conflict misses in multiprocessor applications. It enables applications to make better use of the increased aggregate cache size available in a multiprocessor. This technique uses the compiler's knowledge of the access patterns of the parallelized applications to direct the operating system's virtual memory page mapping strategy. We demonstrate that this technique can lead to significant performance impr ...

23 Fortran 90D/HPF compiler for distributed memory MIMD computers: design, implementation, and performance results 

 Z. Bozkus, A. Choudhary, G. Fox, T. Haupt, S. Ranka
 December 1993 **Proceedings of the 1993 ACM/IEEE conference on Supercomputing**

Publisher: ACM Press

Full text available:  [pdf\(971.18 KB\)](#) Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#)

24 Using dataflow analysis techniques to reduce ownership overhead in cache coherence protocols 

 Jonas Skeppstedt, Per Stenström
 November 1996 **ACM Transactions on Programming Languages and Systems**

(TOPLAS), Volume 18 Issue 6

Publisher: ACM Press

Full text available: [pdf\(284.68 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#), [review](#)

In this article, we explore the potential of classical dataflow analysis techniques in removing overhead in write-invalidate cache coherence protocols for shared-memory multiprocessors. We construct the compiler algorithms with varying degree of sophistication that detect loads followed by stores to the same address. Such loads are marked and constitute a hint to the cache to obtain an exclusive copy of the block so that the subsequent store does not introduce access penalties. The simplest ...

Keywords: cache coherence, dataflow analysis, performance evaluation

25 Improving data locality with loop transformations

 Kathryn S. McKinley, Steve Carr, Chau-Wen Tseng

July 1996 **ACM Transactions on Programming Languages and Systems (TOPLAS)**, Volume 18 Issue 4

Publisher: ACM Press

Full text available: [pdf\(411.40 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#), [review](#)

In the past decade, processor speed has become significantly faster than memory speed. Small, fast cache memories are designed to overcome this discrepancy, but they are only effective when programs exhibit data locality. In this article, we present compiler optimizations to improve data locality based on a simple yet accurate cost model. The model computes both temporal and spatial reuse of cache lines to find desirable loop organizati ...

Keywords: Cache, compiler optimization, data locality, loop distribution, loop fusion, loop permutation, loop reversal, loop transformations, microprocessors, simulation

26 Efficient resolution of sparse indirections in data-parallel compilers

 Manuel Ujaldon, Emilio L. Zapata

July 1995 **Proceedings of the 9th international conference on Supercomputing**

Publisher: ACM Press

Full text available: [pdf\(1.05 MB\)](#) Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#)

27 Cache coherence using local knowledge

 E. Darnell, K. Kennedy

December 1993 **Proceedings of the 1993 ACM/IEEE conference on Supercomputing**

Publisher: ACM Press

Full text available: [pdf\(1.15 MB\)](#) Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#)

28 Unified compilation of Fortran 77D and 90D

 Alok Choudhary, Geoffrey Fox, Seema Hiranandani, Ken Kennedy, Charles Koelbel, Sanjay Ranka, Chau-Wen Tseng

March 1993 **ACM Letters on Programming Languages and Systems (LOPLAS)**, Volume 2 Issue 1-4

Publisher: ACM Press

Full text available: Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index](#)

 pdf(1.29 MB)[terms, review](#)

We present a unified approach to compiling Fortran 77D and Fortran 90D programs for efficient execution of MIMD distributed-memory machines. The integrated Fortran D compiler relies on two key observations. First, array constructs may be scalarized into FORALL loops without loss of information. Second, loop fusion, partitioning, and sectioning optimizations are essential for both Fortran D dialects.

Keywords: Fortran D, parallel languages, parallel programming

29 [Graph contraction for physical optimization methods: a quality-cost tradeoff for mapping data on parallel computers](#) 

 N. Mansour, R. Ponnusamy, A. Choudhary, G. C. Fox

August 1993 **Proceedings of the 7th international conference on Supercomputing**

Publisher: ACM Press

Full text available:  pdf(733.23 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

Mapping data to parallel computers aims at minimizing the execution time of the associated application. However, it can take an unacceptable amount of time in comparison with the execution time of the application if the size of the problem is large. In this paper, first we motivate the case for graph contraction as a means for reducing the problem size. We restrict our discussion to applications where the problem domain can be described using a graph (e.g., computational fluid dynamics appl ...

30 [Architecture-independent scientific programming in data parallel C: three case studies](#) 

 Philip J. Hatcher, Michael J. Quinn, Ray J. Anderson, Anthony J. Lapadula, Bradley K. Seevers, Andrew F. Bennett

August 1991 **Proceedings of the 1991 ACM/IEEE conference on Supercomputing**

Publisher: ACM Press

Full text available:  pdf(1.05 MB) Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#)

31 [Cooperative shared memory: software and hardware for scalable multiprocessor](#) 

 Mark D. Hill, James R. Larus, Steven K. Reinhardt, David A. Wood

September 1992 **ACM SIGPLAN Notices , Proceedings of the fifth international conference on Architectural support for programming languages and operating systems ASPLOS-V**, Volume 27 Issue 9

Publisher: ACM Press

Full text available:  pdf(1.35 MB) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

We believe the absence of massively-parallel, shared-memory machines follows from the lack of a shared-memory programming performance model that can inform programmers of the cost of operations (so they can avoid expensive ones) and can tell hardware designers which cases are common (so they can build simple hardware to optimize them). Cooperative shared memory, our approach to shared-memory design, addresses this problem. Our initial implementation of cooperativ ...

32 [An architecture for software-controlled data prefetching](#) 

 Alexander C. Klaiber, Henry M. Levy

April 1991 **ACM SIGARCH Computer Architecture News , Proceedings of the 18th annual international symposium on Computer architecture ISCA '91**, Volume 19 Issue 3

Publisher: ACM Press

Full text available:  pdf(1.16 MB)

Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#)

33 Compiler optimizations for improving data locality 

 Steve Carr, Kathryn S. McKinley, Chau-Wen Tseng

November 1994 **ACM SIGPLAN Notices , ACM SIGOPS Operating Systems Review , Proceedings of the sixth international conference on Architectural support for programming languages and operating systems ASILOPS-VI**, Volume 29 , 28 Issue 11 , 5

Publisher: ACM Press

Full text available:  pdf(1.34 MB)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

In the past decade, processor speed has become significantly faster than memory speed. Small, fast cache memories are designed to overcome this discrepancy, but they are only effective when programs exhibit data locality. In this paper, we present compiler optimizations to improve data locality based on a simple yet accurate cost model. The model computes both temporal and spatial reuse of cache lines to find desirable loop organizations. T ...

34 OpenMP on networks of workstations 

Honghui Lu, Y. Charlie Hu, Willy Zwaenepoel

November 1998 **Proceedings of the 1998 ACM/IEEE conference on Supercomputing (CDROM)**

Publisher: IEEE Computer Society

Full text available:  pdf(202.91 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#)

We describe an implementation of a sizable subset of OpenMP on networks of workstations (NOWs). By extending the availability of OpenMP to NOWs, we overcome one of its primary drawbacks compared to MPI, namely lack of portability to environments other than hardware shared memory machines. In order to support OpenMP execution on NOWs, our compiler targets a software distributed shared memory system (DSM) which provides multi-threaded execution and memory consistency. This paper presents two contri ...

35 Associative and Parallel Processors 

 Kenneth J. Thurber, Leon D. Wald

December 1975 **ACM Computing Surveys (CSUR)**, Volume 7 Issue 4

Publisher: ACM Press

Full text available:  pdf(2.62 MB) Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#)

36 A compiler method for the parallel execution of irregular reductions in scalable shared memory multiprocessors 

 E. Gutiérrez, O. Plata, E. L. Zapata

May 2000 **Proceedings of the 14th international conference on Supercomputing**

Publisher: ACM Press

Full text available:  pdf(898.78 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

This paper presents a new parallelization method for reductions of arrays with subscripted subscripts on scalable shared memory multiprocessors. The mapping of computations is based on grouping reduction loop iterations into sets that are further assigned to the cooperating threads of computation. Iterations belonging to the same set are chosen in

such a way that update different entries in the reduction array. That is, the loop distribution implies a conflict-free write distribution of the ...

37 Evaluation of compiler optimizations for Fortran D on MIMD distributed memory machines



Seema Hiranandani, Ken Kennedy, Chau-Wen Tseng

August 1992 **Proceedings of the 6th international conference on Supercomputing**

Publisher: ACM Press

Full text available:  [pdf\(1.74 MB\)](#)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

The Fortran D compiler uses data decomposition specifications to automatically translate Fortran programs for execution on MIMD distributed-memory machines. This paper introduces and classifies a number of advanced optimizations needed to achieve acceptable performance; they are analyzed and empirically evaluated for stencil computations. Profitability formulas are derived for each optimization. Results show that exploiting parallelism for pipelined computations, reductions, and scans is vi ...

Results 21 - 37 of 37

Result page: [previous](#) [1](#) [2](#)

The ACM Portal is published by the Association for Computing Machinery. Copyright © 2006 ACM, Inc.
[Terms of Usage](#) [Privacy Policy](#) [Code of Ethics](#) [Contact Us](#)

Useful downloads:  [Adobe Acrobat](#)  [QuickTime](#)  [Windows Media Player](#)  [Real Player](#)